

리소스 리스트를 사용한 새로운 분산형 컨퍼런스 구조

장춘서*, 이기수*

A New Distributed Conference Architecture using Resource Lists

Choonseo Jang*, Ky-Soo Lee*

요 약

본 논문에서는 복수개의 컨퍼런스 서버로 구성되는 분산형 컨퍼런스 환경에서 컨퍼런스 참가자들의 URI로 구성된 리소스 리스트를 사용하여 서버와 참가자들 사이에서 교환되는 SIP 메시지 양을 크게 줄이고 아울러 컨퍼런스 서버의 처리 시간 및 부하도 줄일 수 있는 새로운 분산형 컨퍼런스 구조를 제안하였다. 제안된 방안에서는 리소스 리스트를 사용하여 컨퍼런스 참가 시에 소요되는 SIP 메시지 수를 크게 줄일 수 있고 컨퍼런스 이벤트 패키지 등록 및 세션 유지를 위해 필요한 SIP 메시지 수도 크게 줄일 수 있다. 따라서 본 구조를 사용하여 컨퍼런스 시스템의 확장성을 높일 수 있고 네트워크 트래픽을 줄일 수 있게 되며 이를 위하여 본 논문에서는 리소스 리스트를 처리하는데 필요한 요소들이 추가된 새로운 컨퍼런스 정보 데이터 포맷이 제시되었다. 또 본 논문에서는 동적으로 컨퍼런스 서버들이 컨퍼런스 환경에 추가되는 과정이 제시되었고 각 컨퍼런스 서버들과 컨퍼런스 참가자들 사이의 SIP 메시지 및 컨퍼런스 정보 데이터 교환 절차도 함께 제시되었다. 제안된 시스템의 성능은 실험을 통하여 분석하였다.

▶ Keyword : 세션시작프로토콜, 분산 컨퍼런스, 리소스 리스트

Abstract

In this paper, We have proposed a new distributed conference architecture which can reduce largely SIP(Session Initiation Protocol) messages exchanged between servers and participants in distributed conference environment by using resource lists which consist of conference participants URIs. This architecture can also reduce processing time and loads of conference servers. In our proposed method, the number of SIP messages required for participating in the conference can be

• 제1저자 : 장춘서 • 교신저자 : 이기수

• 투고일 : 2011. 06. 27, 심사일 : 2011. 07. 13, 게재확정일 : 2011. 07. 25.

* 금오공과대학교 컴퓨터공학과 교수(Dept. of Computer Engineering, Kumoh. Inst. of Tech.)

※ 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

largely reduced, and the number of SIP messages required for subscription of conference event package and maintaining sessions can also be largely reduced by using resource lists. Therefore expandability of the conference system increases and network traffic can be reduced by using our conference architecture. And We also have proposed in this paper a new conference information data format with some added elements that are necessary for processing resource lists. Furthermore the procedure for dynamically adding conference servers to the conference environment has been suggested, and the exchanging process of SIP messages and conference information data between conference servers and conference participants has been also suggested. The performance of our proposed system has been evaluated by experiments.

▶ Keyword : Session Initiation Protocol, Distributed Conference, Resource Lists

I. 서론

하나의 컨퍼런스 서버를 가지는 중앙 집중형 컨퍼런스 구조[1][2]는 컨퍼런스 참가자 수가 증가 할수록 성능에 제약을 받으므로 이러한 경우 복수개의 컨퍼런스 서버를 가지는 분산형 컨퍼런스 구조가 사용된다.[3][4] 본 논문에서는 이와 같은 분산형 컨퍼런스 구조에서 컨퍼런스 참가자들의 URI로 구성된 리소스 리스트를 사용하여 컨퍼런스 서버들과 컨퍼런스 참가자 사이에서 교환되는 SIP[5] 메시지 양을 크게 줄일 수 있는 새로운 방안을 제안하였다. 이와 같이 SIP 메시지 양을 줄임으로써 컨퍼런스 서버에서의 메시지 처리에 소요되는 부하도 줄일 수 있어 시스템의 확장성을 높일 수 있게 된다. 또 네트워크 트래픽도 줄일 수 있어 특히 통신 대역폭의 제한을 받는 모바일 디바이스 경우나 무선 네트워크 환경에서 유용하게 사용될 수 있다.

본 논문에서 제안한 컨퍼런스 리소스 리스트는 컨퍼런스 참가자들의 URI로 구성되며 이를 사용하여 컨퍼런스 참가시에 소요되는 SIP 메시지 수를 크게 줄일 수 있다. 주 컨퍼런스 서버는 리소스 리스트 서버로 동작하여 컨퍼런스 참가자로부터 SIP INVITE 메시지를 받으면 내부의 리소스 리스트 정보가 들어있는 데이터베이스를 사용하여 각 참가자들에 대한 SIP REFER 메시지를 사용하지 않고도 해당 참가자들이 참가하는 컨퍼런스를 생성할 수 있게 된다. 또 컨퍼런스 이벤트 패키지 등록 시에도 이와 같은 리소스 리스트를 사용하여 필요한 SUBSCRIBE 메시지 수를 크게 줄일 수 있고 컨퍼런스 이벤트 패키지 등록 유지에 사용되는 리프레시도 효율적으로 처리 할 수 있게 된다.

이와 같은 기능을 위하여 본 논문에서는 새로운 컨퍼런스 정보 데이터 포맷이 제시되었으며 이 포맷에는 리소스 리스

트를 처리 할 수 있는 필요 요소들이 추가되었다. 또 리소스 리스트를 사용하는 환경에서 동적으로 컨퍼런스 서버들이 컨퍼런스 환경에 추가되는 과정이 제시되었고 주 컨퍼런스 서버 및 각 부 컨퍼런스 서버들과 컨퍼런스 참가자들 사이의 SIP 메시지 및 컨퍼런스 정보 데이터 교환 절차도 함께 제시되었다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 기본적인 분산형 컨퍼런스 구조를 설명하고 컨퍼런스 정보를 각 참가자들에게 제공하기위한 컨퍼런스 이벤트 패키지에 대해 설명한다. III장에서는 본 논문에서 제안하는 리소스 리스트를 사용하는 새로운 분산형 컨퍼런스 구조 및 필요 요소들이 새롭게 추가된 컨퍼런스 정보 데이터 포맷에 대한 설계와 구현에 대해 설명한다. 또 컨퍼런스 서버들과 참가자들 사이의 SIP 메시지 및 컨퍼런스 정보 데이터 교환 절차도 제시되었다. IV장에서는 실험을 통하여 본 논문에서 제안한 시스템에 대한 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

1. 기본적 분산형 컨퍼런스 구조

그림 1의 기본적 분산형 컨퍼런스 구조는 하나의 컨퍼런스 서버가 컨퍼런스 전체를 제어하고 관리하는 중앙 집중형 컨퍼런스 구조 대신 복수개의 컨퍼런스 서버들로 구성된다. 각 컨퍼런스 서버는 포커스와 믹서를 갖추고 있으며 참가자들을 서로 나누어 처리함으로써 컨퍼런스의 확장성을 높일 수 있다.

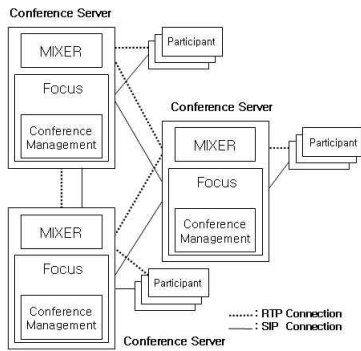


그림 1. 분산형 컨퍼런스 구조
Fig. 1 Distributed Conference Architecture

여기서 컨퍼런스 포커스는 컨퍼런스 세션의 설정 등 컨퍼런스와 관련된 각종 제어 기능을 제공하며 이들 동작은 각종 SIP 메시지 기반으로 이루어진다.[6] 믹서는 참가자와 서버 사이에 연결된 RTP(Real-Time Transport Protocol)[7] 세션을 사용하여 미디어 패킷 스트림을 조합하고 전체 참가자들에게 분배하는 기능을 한다. 또 컨퍼런스 환경에서 공유 자원에 대한 접근을 제어하기 위하여 플로어 제어 기능이 사용되기도 한다.

그러나 이와 같은 분산형 컨퍼런스 구조는 컨퍼런스 참가자가 증가할 경우 동적으로 새로운 컨퍼런스 서버가 추가되도록 제어되어야 하고 추가된 컨퍼런스 서버와 기존 서버들 사이의 연동 및 컨퍼런스 참가자의 분배와 이에 따르는 SIP 세션 연결을 위한 메시지 교환 절차 등 여러 문제들이 있다.[8] 본 논문에서는 분산형 컨퍼런스 구조에서는 새롭게 시도되는 컨퍼런스 참가자들의 URI로 구성된 리소스 리스트를 사용하여 컨퍼런스 서버들과 컨퍼런스 참가자들 사이에서 교환되는 SIP 메시지 양을 크게 줄일 수 있는 새로운 방안을 제안하였다.

2. 컨퍼런스 이벤트 패키지와 컨퍼런스 정보 데이터

컨퍼런스 이벤트 패키지[9][10]는 변화하는 컨퍼런스 정보를 컨퍼런스 참가자들에게 제공하는 기능을 하며 이를 위해서는 자신을 관리하는 컨퍼런스 서버에게 SIP SUBSCRIBE 메시지를 보내어 등록을 한다. 컨퍼런스 서버는 컨퍼런스 참가자가 추가되거나 탈퇴하는 등 컨퍼런스 정보의 변화가 생기면 SIP NOTIFY 메시지를 사용해 각 컨퍼런스 참가자들에게 이를 통지한다.

통지 시 사용되는 컨퍼런스 정보 데이터는 XML 문서인 application/conference-info+xml 포맷을 가지며 SIP NOTIFY

Y 메시지의 헤더 이후 부분에 넣어서 전송된다. 이 컨퍼런스 정보 데이터 포맷은 'conference-info'를 최상위 요소(element)로 가지며 하위 요소(child element)로써 'conference-description'이 있고 이는 컨퍼런스 전체에 대한 정보를 기술한다. 이 밖의 하위 요소로는 컨퍼런스를 주관하는 호스트에 대한 정보를 나타내는 'host-info'와 현재 컨퍼런스 상태를 표시하기 위한 'conference-state' 및 컨퍼런스 참가자에 대한 개별 정보를 나타내기 위한 'users' 등이 있다. 이들 요소들은 모두 각각의 기능을 가지는 하위 요소들을 가진다.

요소 'users'는 하위 요소로 각 참가자들의 정보를 나타내는 'user'를 가지며 이 요소는 속성으로 각 참가자를 구분하기 위한 'entity'를 가진다. 요소 'user'의 중요한 하위 요소로는 'endpoint'가 있고 이는 각 참가자들이 컨퍼런스에 참여하면서 사용하는 디바이스 및 SIP 신호 세션에 대한 정보를 나타낸다. 이 요소는 다시 컨퍼런스에 참가한 방식을 나타내는 'joining-method'와 컨퍼런스 서버와의 미디어 스트림 정보를 나타내는 'media' 및 디바이스의 연결 상태를 나타내는 'status' 등의 여러 하위 요소를 가진다. 요소 'media'는 컨퍼런스 내의 여러 미디어 스트림 중 해당 스트림을 구분하기 위해서 포커스에 의해 부여되는 'label'과 실제 소스 미디어를 나타내는 'src-id' 등의 여러 하위 요소를 가진다.

그러나 분산형 컨퍼런스 환경에서 리소스 리스트를 사용하여 동적으로 컨퍼런스 서버들을 추가하고 컨퍼런스 참가자들 서버들 사이에서 고르게 분배하는 데 필요한 컨퍼런스 정보 데이터 포맷 및 컨퍼런스 정보 교환 절차가 아직 제안되고 있지 않다. 따라서 본 논문에서는 리소스 리스트를 사용하는 분산형 컨퍼런스 구조에서 사용할 수 있는 새로운 컨퍼런스 정보 데이터 포맷을 제안하였다.

III. 시스템 설계 및 구현

1. 리소스 리스트를 사용하는 분산형 컨퍼런스 구조 설계

본 논문에서 설계한 리소스 리스트를 사용한 분산형 컨퍼런스 구조는 하나의 주 컨퍼런스 서버와 복수 개의 부 컨퍼런스 서버로 구성된다. 주 컨퍼런스 서버는 컨퍼런스 참가자와 컨퍼런스 서버 사이의 컨퍼런스 세션을 생성하고 관리하는 포커스, 미디어 패킷 스트림을 서로 교환하고 분배하는 기능을 하는 믹서, 리소스 리스트를 처리하는 리소스 리스트 제어 모듈 및 컨퍼런스 정보 데이터베이스로 구성된다. 부

컨퍼런스 서버는 컨퍼런스 정보 데이터베이스를 제외하면 주 컨퍼런스 서버와 동일한 구성을 가진다. 그림 2에 이를 보였다.

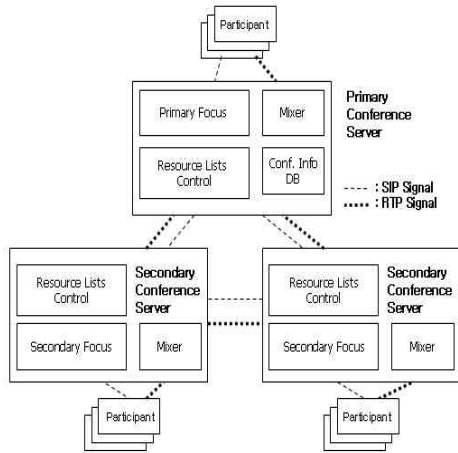


그림 2. 설계된 리소스 리스트를 사용하는 분산형 컨퍼런스 구조
 Fig.2. Designed Architecture of Distributed Conference using Resource Lists

본 연구에서는 컨퍼런스 서버와 참가자 사이에 리소스 리스트 지원 여부를 확인하기 위해 SIP Supported 헤더와 SIP Require 헤더를 사용하였다. 리소스 리스트를 지원하는 참가자는 SIP INVITE 메시지의 Supported 헤더에 'conflist' 태그를 포함시키고 이 메시지를 받은 컨퍼런스 서버는 이에 대한 응답 메시지의 Require 헤더에 동일한 태그를 넣어 보내도록 설계하였다. 컨퍼런스 참가자가 주 컨퍼런스 서버의 주 포커스에게 SIP INVITE 메시지를 보내면 리소스 리스트 제어 모듈은 본 논문에서 제안된 컨퍼런스 정보 데이터 포맷으로 구축된 컨퍼런스 정보 데이터베이스에서 해당 참가자에 대한 컨퍼런스 리소스 리스트를 추출하여 해당 참가자에게 직접 SIP INVITE 메시지를 보내어 컨퍼런스에 참가시킨다. 이 과정에서 SIP REFER 메시지 교환 절차가 필요 없게 되어 SIP 메시지 수를 크게 줄일 수 있게 된다.

이때 컨퍼런스 참가자 수가 미리 정해진 값보다 커질 경우 새로운 컨퍼런스 서버가 동적으로 추가되고 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에 컨퍼런스 동작 및 관리에 필요한 컨퍼런스 정보가 교환되며 또 참가자 수를 각 컨퍼런스 서버에게 균등하게 분배하는 과정이 이루어진다. 이와 같은 컨퍼런스 동작 과정에 대한 순서도를 그림 3에 보였다.

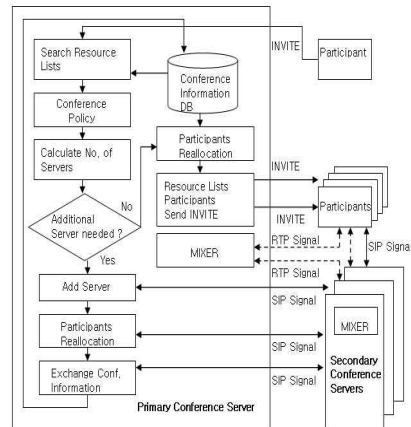


그림 3 리소스 리스트를 사용한 컨퍼런스 동작 과정 순서도
 Fig. 3 Flowchart of Conference Operation using Resource Lists

그림 3에서는 새로운 참가자가 컨퍼런스 URI를 담은 SIP INVITE 메시지를 주 컨퍼런스 서버에게 보내면 주 컨퍼런스 서버는 SIP Contact 헤더 정보로부터 컨퍼런스 정보 데이터베이스를 액세스하여 해당 컨퍼런스 리소스 리스트를 얻는다. 다음 참가자 수에 대한 컨퍼런스 정책을 사용하여 컨퍼런스 리소스 리스트의 참가자들을 처리하는데 필요한 컨퍼런스 서버의 개수를 계산한다. 현재 동작 중인 컨퍼런스 서버들 이외의 새로운 컨퍼런스 서버가 필요한 경우 데이터베이스에서 선택된 컨퍼런스 서버에게 INVITE 메시지를 보내고 RTP 연결을 맺어 부 컨퍼런스 서버를 추가한다. 새로운 컨퍼런스 서버 추가 후 각 컨퍼런스 서버의 부하에 대한 균등 분배를 위해 전체 컨퍼런스 서버들에 대한 참가자 재분배 과정이 행해지며 이때 컨퍼런스 리소스 리스트의 참가자들에 대한 처리도 각 컨퍼런스 서버들로 분배된다. 다음 리소스 리스트 정보를 포함하여 각 서버들에대한 컨퍼런스 정보 데이터 교환이 행해진다.

컨퍼런스 정책에 의해 기존 컨퍼런스 서버들로 처리가 가능하다고 판단된 경우 참가자 재분배 과정을 거쳐 각 컨퍼런스 서버 별로 리소스 리스트의 참가자들에 대해 SIP INVITE 메시지가 보내져 컨퍼런스에 참가시킨다. 이와 같이 리소스 리스트를 사용하여 별도의 SIP REFER 메시지 없이 컨퍼런스가 생성될 수 있어 시스템 내에서 교환되는 SIP 메시지 양을 크게 줄일 수 있으며 참가자 수에 따라 동적으로 컨퍼런스 서버가 추가될 수 있다.

2. 리소스 리스트 컨퍼런스 정보 데이터 포맷 및 컨퍼런스 정보 교환 절차

본 연구에서는 리소스 리스트를 처리하기 위하여 다음과 같은 필요 요소들이 추가된 새로운 컨퍼런스 정보 데이터 포맷이 설계되었다. 먼저 최상위 요소 'conference-info'의 하위 요소로 새롭게 'conf-rsc-list'가 추가되었다. 이 요소는 컨퍼런스 리소스 리스트를 나타내며 속성으로 각 컨퍼런스 리소스 리스트를 구분하기 위한 'rsc-id'를 가진다. 요소 'conf-rsc-list'의 하위 요소로는 각 컨퍼런스 리소스를 나타내는 'conf-resource'가 온다. 이 요소의 속성으로는 각 컨퍼런스 참가자의 SIP URI를 나타내는 'user-uri'가 있고 하위 요소로는 리소스 상태를 표시하기 위한 'instance'를 두었다.

컨퍼런스 참가자는 컨퍼런스 서버에게 보내는 INVITE 메시지의 Contact 헤더에 파라미터 'user-rsc-id'를 사용하여 리소스 리스트를 나타낸다. 현재 동작 중인 전체 컨퍼런스 서버의 개수를 나타내는 'conf-servers'와 각 컨퍼런스 서버의 주요 항목을 표시하기 위한 'svr-description'이 컨퍼런스 전체에 대한 정보를 기술하는 요소인 'conference-description'의 하위 요소로써 추가되었다. 이중 'svr-description'은 하위요소로 각 컨퍼런스 서버의 최대 참가자 처리 용량인 'max-capacity'와 각각의 컨퍼런스 서버를 구분하기 위한 'label'을 가진다.

다음 각 컨퍼런스 서버에 현재 연결된 참가자 목록인 'users-list'와 전체 컨퍼런스 참가자의 수를 나타내기 위한 'total-users'가 현재 컨퍼런스의 상태를 표시하기 위한 요소인 'conference-state'의 하위요소로 추가되었다. 요소 'users-list'는 하위 요소로 각각의 컨퍼런스 참가자를 구분하기 위한 'user-identity'와 각 컨퍼런스 서버를 구분하기 위한 'server-ID'를 가진다. 각 컨퍼런스 참가자들을 나타내는 요소 'user'에는 하위 요소로 'suppress'가 추가되었고 이의 속성 'allow'가 TRUE 값을 가지면 세션 연장을 위한 리프레시 동작에서 SIP SUBSCRIBE 메시지에 대한 NOTIFY 통지 메시지가 생략되어 전체적인 SIP 메시지 발생 양을 더욱 줄일 수 있게 된다.

이와 같은 리소스 리스트를 처리하기 위한 필요 요소들이 새롭게 추가된 컨퍼런스 정보 데이터 포맷을 사용한 컨퍼런스 서버들과 참가자들 사이의 컨퍼런스 정보 교환 절차를 그림 4에 보였다. 여기서 컨퍼런스 참가자가 주 컨퍼런스 서버에게 INVITE 메시지를 보내면 주 컨퍼런스 서버는 컨퍼런스 정책에 의해 컨퍼런스 서버를 추가하는 과정으로 들어간다. 주 컨퍼런스 서버는 부 컨퍼런스 서버(Secondary Server B)에게 INVITE 메시지를 보내고 RTP 연결을 맺은 후 SUBSCRIBE와 NOTIFY 메시지를 사용해 컨퍼런스 정보를

교환한다. 다음 해당 참가자에게 새롭게 추가된 부 컨퍼런스 서버의 주소를 담은 SIP 302 Redirection 메시지를 보내고 참가자는 이 서버에게 INVITE 메시지를 보내고 RTP 연결을 맺는다.

새롭게 추가된 부 컨퍼런스 서버는 컨퍼런스 리소스 리스트를 사용하여 해당 SIP 주소로 INVITE 메시지를 보내어 참가자를 컨퍼런스에 참가시키고 RTP 세션을 맺어 미디어 스트림 처리를 담당하게 된다. 이 컨퍼런스 서버는 컨퍼런스 참가자들에게 NOTIFY 메시지로 컨퍼런스 정보를 통지하는데 이때 리소스 리스트를 사용하여 각 참가자들에 대한 컨퍼런스 정보 이벤트 패키지를 자체적으로 등록시키므로 SUBSCRIBE 메시지의 사용량을 줄일 수 있게 된다. 새롭게 추가된 부 컨퍼런스 서버는 리소스 리스트의 참가자들을 모두 컨퍼런스에 참가 시킨 후 변경된 컨퍼런스 정보를 주 컨퍼런스 서버에게 NOTIFY 메시지를 보내어 통지하며 주 컨퍼런스 서버는 이를 다른 부 컨퍼런스 서버들에게도 알리고 자신이 담당하는 참가자들에게도 알린다. 각 부 컨퍼런스 서버들은 이와 같은 컨퍼런스 정보를 자신이 담당하는 참가자들에게도 알림으로써 참가자 전체가 새로운 컨퍼런스 정보를 공유 할 수 있게 된다.

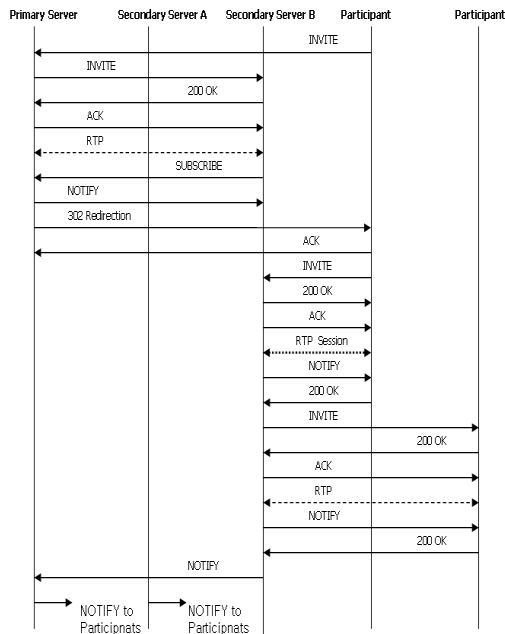


그림 4. SIP 메시지 및 컨퍼런스 정보 교환 절차
 Fig. 4 Exchange Procedure of SIP Messages and Conference Information

IV. 성능 분석

본 논문에서 제안한 리소스 리스트를 사용한 새로운 분산형 컨퍼런스 구조의 성능 분석을 위하여 컨퍼런스 서버로는 커널 버전 2.6인 리눅스가 설치된 두 대의 PC가 사용되었고 이중 한 대는 주 컨퍼런스 서버이고 한대는 부 컨퍼런스 서버이다. 컨퍼런스 참가자용 PC로는 MS 윈도우즈 XP가 설치된 15대의 PC가 사용되었다. 각 PC는 1.8GHz 코어2듀오 CPU와 2GB 메인메모리를 가지고 있다. 각 PC의 네트워크 인터페이스를 포함한 LAN 환경의 속도는 100Mbps로 하였고 하나의 LAN 세그먼트로 구성하였다.

먼저 본 논문에서 제안한 컨퍼런스 리소스 리스트를 INVITE 메시지 처리에 사용한 경우와 기존 방식을 사용한 경우의 서버에서의 메시지 처리 시간을 비교하였다. 분석을 위하여 컨퍼런스 참가자 수 6명까지는 주 컨퍼런스 서버만 사용하여 동작하며 이를 초과 시 부 컨퍼런스 서버가 추가되도록 컨퍼런스 정책을 설정하였다. 그림 5는 측정된 결과이다.

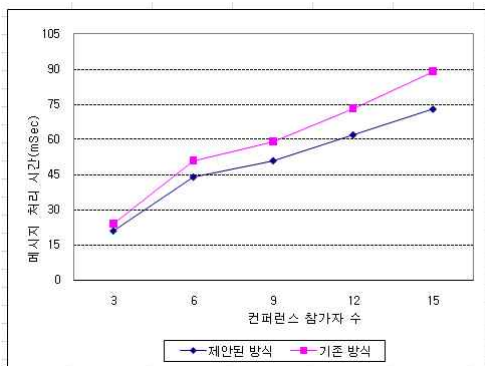


그림 5. 리소스 리스트를 사용한 INVITE 메시지 처리시간 비교
Fig. 5 Comparison of INVITE Message Processing Time using Resource Lists

주 컨퍼런스 서버만 동작하는 경우인 참가자 수가 3명과 6명 일 때의 측정 결과는 제안된 리소스 리스트를 사용한 경우 메시지 처리 시간이 12.5%와 13.7% 감소함을 보여주고 있다. 참가자 수가 6명을 초과하여 두 개의 컨퍼런스 서버가 동작하는 경우인 9명, 12명, 15명 일 때는 각각 13.5%, 15.0%, 17.9%가 감소되어 참가자 수가 증가 할수록 리소스 리스트를 사용한 INVITE 메시지 처리 시간의 감소 효과가 점차 커짐을 알 수 있다.

그림 6은 리소스 리스트를 SUBSCRIBE 메시지 처리에 사용한 경우와 기존 방식을 사용한 경우의 서버에서의 메시

지 처리 시간을 비교 측정된 결과이다.

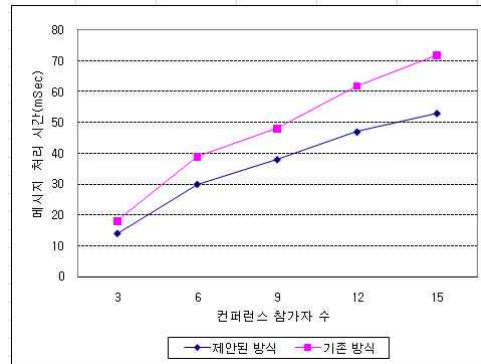


그림 6. 리소스 리스트를 사용한 SUBSCRIBE 메시지 처리시간 비교
Fig. 6 Comparison of SUBSCRIBE Message Processing Time using Resource Lists

여기서 참가자 수가 3명과 6명 일 때의 측정 결과는 제안된 리소스 리스트를 사용한 경우 메시지 처리 시간이 22.2%와 23.1% 감소함을 보여주고 있고 이때는 주 컨퍼런스 서버만 동작하는 경우이다. 참가자 수가 9명, 12명, 15명 인 경우는 두 개의 컨퍼런스 서버가 동작하며 이때는 각각 22.9%, 24.2%, 26.3%가 감소되어 참가자 수가 증가 할수록 리소스 리스트를 사용한 SUBSCRIBE 메시지 처리 시간의 감소 효과가 점차 커짐을 알 수 있다.

V. 결론 및 향후 과제

본 논문에서는 분산형 컨퍼런스 환경에서 리소스 리스트를 사용하여 서버와 참가자들 사이에서 교환되는 SIP 메시지 양을 크게 줄이고 아울러 컨퍼런스 서버의 처리 시간 및 부하도 줄일 수 있는 새로운 분산형 컨퍼런스 구조를 제안하였다. 이를 위하여 필요 요소들이 추가된 새로운 컨퍼런스 정보 데이터 포맷이 제시되었고 리소스 리스트를 사용하는 환경에서 동적으로 새로운 컨퍼런스 서버들이 컨퍼런스 환경에 추가되는 과정과 컨퍼런스 서버들 및 컨퍼런스 참가자들 사이의 SIP 메시지 및 컨퍼런스 정보 데이터 교환 절차도 함께 제시되었다.

컨퍼런스 참가자들의 URI로 구성되는 리소스 리스트를 사용하여 컨퍼런스 참가 시 및 컨퍼런스 이벤트 패키지 등록 시에 소요되는 SIP 메시지 수를 크게 줄일 수 있으며 따라서 컨퍼런스 시스템의 확장성을 높일 수 있고 네트워크 트래픽을 줄일 수 있게 된다. 제안된 시스템의 성능은 실험을 통하여 분석하였으며 컨퍼런스 참가자 수가 증가 할수록 리

소스 리스트를 사용한 메시지 처리 시간의 감소 효과가 커짐을 확인할 수 있었다. 향후 과제로는 컨퍼런스 참가자 수를 현재보다 늘리고 무선 모바일 환경을 구축하여 제안된 시스템이 무선 모바일 환경에서 얼마만큼 효율적인가를 관측하고 이에 맞춘 개선된 방식을 연구할 필요가 있다.

참고 문헌

- [1] M. Barnes, C. Boulton and O. Levin, "A Framework for Centralized Conferencing," RFC 5239, June. 2008.
- [2] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol (SIP)," RFC 4353, Feb. 2006.
- [3] C. S. Jang, H. K. Cho, K. S. Lee, "A Method of Efficient Conference Event Package Processing in Distributed Conference Environment" Journal of The Korea Society of Computer and Information, Vol 13, No.7, pp 199-205, Dec. 2008.
- [4] Y. Cho et al., "Distributed management architecture for multimedia conferencing using SIP," Int. Conf. DFMA, pp.98-105, Feb. 2006.
- [5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "Session Initiation Protocol," RFC 3261, June 2002.
- [6] C. S. Jang, H. K. Cho, K. S. Lee, "A Method of Efficient SIP Messages Processing for Conference Focus" Journal of The Korea Society of Computer and Information, Vol. 12, No. 6, pp. 187-192, Dec. 2007.
- [7] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2005.
- [8] Y. H. Cho, M. S. Jeong, "Policy-based distributed management architecture for large-scale enterprise conferencing service using SIP," IEEE Journal on Communications, pp.1934-1949, Oct. 2005.
- [9] J. Rosenberg, H. Schulzrinne and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State," RFC 4575, August 2006.
- [10] G. Camarillo, et al., "Conference Event Package Data Format Extension for Centralized Conferencing," Internet-draft, September 2008.

저자 소개



장 춘 서

1983년 2월 : 한국과학기술원 공학 박사

1981년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 교수

관심분야 : SIP, 임베디드 시스템, 인터넷텔레포니

Email : csjang@kumoh.ac.kr



이 기 수

1982년 2월 : 서울대학교 대학원 공학석사

1982년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 교수

관심분야 : SIP, 디지털시스템, 데이터베이스

Email : kyssoolee@kumoh.ac.kr